A Heuristic for Time Constrained Arc Routing Problems

*Richard Eglese**

* Department of Management Science, Lancaster University
Lancaster, Lancashire, LA1 4YX, UK
Email: R.Eglese@lancaster.ac.uk

1 Introduction

Arc routing problems arise, for example, when roads require treatment or customers must be served who are located along roads. So practical examples include the routing of people or vehicles for postal deliveries, meter reading and winter gritting, where the roads need to be treated with salt or some other agent to prevent the surface becoming icy. An arc routing problem is represented by a graph where a subset of the arcs must be treated on one or more routes, possibly subject to constraints.

The paper first describes how the problems tackled have been formulated. The design features of the heuristic algorithm named LOCSAR are described. LOCSAR is based on a tabu search approach and some computational results will be reported.

2 Formulation

The problem is formulated in the following way. Given a connected graph consisting of nodes and undirected arcs or edges of which a subset are required to be treated, the problem is to find a set of vehicle routes which will minimise an associated objective function. Each vehicle route must start and end at a depot and must specify an ordering for the arcs included on the route and whether they are treated on that route or not. Each vehicle has a capacity limit Q and the speed of the vehicle may be different when treating an arc compared to when it is travelling along an arc without treating it, which is known as deadheading. There is also a time limit T by which each required arc must be treated. The objective to be minimised is the sum of four components, each of which may be given its own weighting. The four components are the number of vehicle routes, the total distance travelled, the total of the distances by which the routes exceed the capacity limit Q and the total of the times by which the routes exceed the time deadline T.

An important feature of this formulation is that the two constraints on vehicle capacity and the time deadline are effectively being treated as soft constraints. Solutions are allowed where the vehicle capacity or the time constraint may be exceeded in some cases, but the weights in the objective need to be chosen so that the results are acceptable in practice.

3 Description of LOCSAR

The heuristic algorithm presented here is named LOCSAR (LOCal Search for Arc Routing). LOCSAR is based on examining moves from a local neighbourhood of the current solution to search for an improved solution and the current version incorporates features of Tabu Search to guide the search process.

To define a neighbourhood move in LOCSAR, a section is defined as a sequence of required arcs which are treated consecutively on a route. The length of a section is the number of treated arcs in the section. The arcs do not have to be adjacent, so for a vehicle to travel from the beginning to the end of a section, it may be necessary to deadhead along the shortest path between successive arcs which are being treated, but it will always start and finish with a treated arc.

A move consists of a section being removed from a route and either forming a new route or being inserted into an existing route. The section may be travelled in the original direction or after being reversed.

For small problems, all neighbourhood moves can be considered and evaluated, but for large problems, it may be worthwhile restricting the candidate set of moves to exclude those which are unlikely to lead to low cost solutions. This has been implemented by including a parameter NBDDIST1. If after evaluating the change in deadheading distance resulting from a move, the extra deadheading is more than NBDDIST1, the program ascribes an artificially high cost to this move and does not carry out the remaining calculations to update the change to the objective. This appears to give a worthwhile speed-up without sacrificing solution quality providing NBDDIST1 is not too small.

In the original version of LOCSAR every move is evaluated using every section with length less than or equal to MSL (Maximum Section Length), a parameter set by the user. In addition the method also evaluates moves where a section consists of an entire route from the first treated arc to the last treated arc. This means that the effect of merging any two routes or reversing a route is always evaluated. Tests were carried out with MSL set at 1, 2 and 3 (previous tests showed that moves with MSL greater than 3 were rarely accepted). The results showed that the best solutions did not come consistently from any single value for MSL and the computing time required increased with the value of MSL.

It was therefore decided to introduce a variable neighbourhood size to the algorithm. Initially MSL is set at 1. A new parameter, FIX, is given a value by the user. After FIX iterations, if the best solution has not been improved, MSL is increased by 1 up to a maximum of 3. When a new best solution is found, MSL is reset to 1.

Thus a larger neighbourhood is explored when the algorithm has spent a significant time without improving the best solution.

Candidate moves are held in a table of best neighbourhood moves between routes so that the next best move can be found efficiently subject to tabu restrictions. An aspiration level is set to override the tabu restriction when a sufficiently good move is found and a long term memory element is introduced to encourage diversification.

4 Computational results

Computational results will be reported for various sets of data relating to real road networks. The results show how the performance of LOCSAR is affected by the time deadline (which affects the number of vehicle routes).

These results demonstrate how LOCSAR is able to improve the results from a simple constructive heuristic. Especially for the larger networks, there is a tendency for the computing time to increase as

the number of routes decreases. This is chiefly due to the way that the table of best neighbourhood moves between routes is updated. The number of possibilities which need to be tested increases with the number of arcs treated on each route. When a solution consists of several routes, each one contains relatively few arcs.

LOCSAR has been written so that it will naturally deal with cases where more than one depot is available from which vehicles may start and finish their routes. Each route must start and finish at the same depot. When considering a move which creates a new route, the route is assigned to the depot nearest the first arc to be treated. There is no need to carry out any prior allocation of arcs to depots from which they will be treated. This is particularly useful in cases where the least cost solutions use different numbers of vehicles at each depot.

Results will be given to ilustrate the multi-depot case. These show how the number of vehicles required and the distance travelled can be affected by the number and location of the depots. This can be an important trade-off in practice when management are considering the best number and location for depot sites.

References

[1] Dror M. (editor) (2000) Arc Routing: Theory, Solutions and Applications. Kluwer, Boston.

[2] Eglese R.W. & Li L.Y.O. (1996) A tabu search based heuristic for arc routing with a capacity constraint and time deadline. In I.H. Osman & J.P. Kelly (Eds.) Metaheuristics: Theory and Applications. Kluwer, Boston, pp.633-650.

[3] Li L.Y.O. & Eglese R.W. (1996) An interactive algorithm for vehicle routeing for winter gritting. J. Opl Res. Soc., 47, 217-228.